Exame Preliminar para o Doutorado
Prova de Análise de Algoritmos

IME–USP, 5 de março de 2009
das 14 às 18:30 horas

Instruções:

(i) O candidato pode resolver todas as questões. A banca considerará questões cujos valores somem até 10 pontos de modo que a soma total das notas obtidas seja máxima.

(ii) Todas as soluções **devem** conter justificativas das respostas.

(iii) O candidato deve mencionar claramente os teoremas e propriedades usados para justificar suas afirmações.

(iv) Esta prova contém 4 questões de 1 ponto e 5 de 2 pontos.

(v) Nas questões em que são pedidos que os algoritmos sejam escritos em pseudocódigo, o aluno pode alternativamente usar uma linguagem de programação de sua preferência.

(vi) Comece a responder cada questão em uma folha nova. Não escreva no verso das folhas. Numere as folhas.

Questão 1 [1 ponto] Prove que

$$\sum_{i=1}^{n}\sum_{j=1}^{i} j = \Theta(n^3).$$

Questão 2 [1 ponto] Queremos manter uma fila de prioridades com itens cujas prioridades são dadas pelas suas chaves, que são inteiros não-negativos menores ou iguais a 2009. (Quanto menor a chave, maior a prioridade; desempates podem ser feitos de forma arbitrária.)

As operações que executaremos sobre a nossa fila serão as operações de:
(a) *inicialização* da fila (como a fila vazia);
(b) *inserção* de um item com uma dada chave;
(c) *extração* de um item de máxima prioridade.

Descreva como podemos implementar tal fila de forma que cada uma das operações acima possa ser executada em tempo constante. Escreva em pseudocódigo e faça a análise de complexidade.

Questão 3 [1 ponto] Considere a seguinte sequência de comandos de atribuição:

$$\begin{aligned} x &\leftarrow x + y \\ y &\leftarrow x - y \\ x &\leftarrow x - y \end{aligned}$$

(a) Mostre que se as variáveis x e y contem números inteiros, ao fim da sequência seus valores foram trocados.

(b) Na linguagem C existe o tipo `unsigned int`, cujos valores estão entre 0 e `UINT_MAX` (uma constante cujo valor preciso é irrelevante aqui), com os quais as operações aritméticas são realizadas módulo `UINT_MAX+1`. Mostre que nesse caso o código acima ainda funciona para trocar o valor de duas variáveis.

Questão 4 [1 ponto] Neste problema, fotos e imagens são representadas por matrizes binárias (matrizes de 0's e 1's). Dada uma imagem de um elemento procurado X (uma matriz binária de tamanho $p \times q$) deseja-se procurá-la numa foto F de tamanho $m \times n$. Cada posição de uma das matrizes é também chamada de *pixel*.

(a) Escreva em pseudocódigo uma função de nome PROCURA-SE, especificada de forma a receber parâmetros X, p, q, F, m e n como descritos acima. A função deve devolver VERDADEIRO se existir uma submatriz de F delimitada por p linhas consecutivas e q colunas consecutivas que seja idêntica à matriz X, e FALSO caso contrário.

(b) Em função dos parâmetros fornecidos à função PROCURA-SE, calcule o número de comparações de pixels feitas, no pior caso.

(c) Suponha que haja $N = 10$ milhões de fotos num banco de dados a ser vasculhado. Adote os valores $p = q = 100$, $m = n = 1000$, e suponha que cada comparação de pixels (elementos das matrizes) gasta ao menos 10^{-13} dias (8.64 nanosegundos). Calcule em dias o tempo que um programa que chame a sua função PROCURA-SE N vezes levará, no pior caso, para executar todas as comparações de pixels necessárias na procura de uma imagem X nas N fotos F desse banco de dados.

Questão 5 [2 pontos] Considere o seguinte algoritmo recursivo, cujo argumento n é um inteiro positivo.

ASTERISCO(n)
se $n > 0$
    então ASTERISCO($n - 1$)
        para $i \leftarrow 1$ até n faça
            imprima "$*$"
        ASTERISCO($n - 1$)

Para um dado valor de n, quantos asteriscos serão impressos em uma chamada de ASTERISCO(n)? Justifique a sua resposta mostrando os cálculos que fez para chegar a ela.

Questão 6 [2 pontos] [CLRS 17.1-3] Uma seqüência de n operações é executada sobre uma certa estrutura de dados. Suponha que a i-ésima operação custa

$$\begin{array}{ll} i & \text{se } i \text{ é uma potência de 2,} \\ 1 & \text{em caso contrário.} \end{array}$$

Mostre que o custo amortizado de cada operação é O(1).

Questão 7 [2 pontos] Considere o seguinte algoritmo que determina o maior elemento de um vetor $v[1\,..\,n]$ com $n \geq 2$ números positivos distintos.

```
MÁXIMO(v, n)
1  maior ← 0
2  para i ← 1 até n faça
3      se v[i] > maior
4          então maior ← v[i]
5  devolva maior
```

Suponha que v é uma permutação de 1 a n escolhida ao acaso dentre todas as permutações de 1 a n, de acordo com a distribuição uniforme de probabilidade. Seja X o número de vezes que a variável *maior* é alterada (ou seja, o número de execuções das linha 4 do algoritmo) numa chamada de MÁXIMO(v, n). Note que X é uma variável aleatória. Calcule o valor esperado de X. Deduza depois a ordem de crescimento de $E[X]$ em função de n.

Questão 8 [2 pontos]

(a) Explique informalmente o que significa um problema ser NP-completo. Do ponto de vista prático, qual é a relevância de se determinar que um certo problema é NP-completo?

(b) Para cada uma das afirmações abaixo, diga se ela é *verdadeira*, *falsa*, *verdadeira se* $P \neq NP$ ou *falsa se* $P \neq NP$. Nos dois últimos itens, suponha que A e B são problemas em NP.

1. Não há problemas em P que são NP-completos.
2. Existe apenas algoritmo exponencial para o problema da parada.
3. Existem problemas em P que estão em NP.
4. Se A pode ser polinomialmente reduzido a B e B está em P então A está em P.
5. Se A pode ser polinomialmente reduzido a B e B é NP-completo então A é NP-completo.

Dê uma justificativa **curta** para cada resposta.

Questão 9 [2 pontos] [Mochila 0-1. CLRS 16.2-2] O problema da mochila 0-1 consiste no seguinte: dados números inteiros não-negativos $v_1, \ldots, v_n$, $w_1, \ldots, w_n$ e W, queremos encontrar um subconjunto K de $\{1, \ldots, n\}$ que:

$$\begin{array}{l} \text{satisfaça } \sum_{k \in K} w_k \leq W \text{ e;} \\ \text{maximize } \sum_{k \in K} v_k. \end{array}$$

(Imagine que w_i é o *peso* e v_i é o *valor* do objeto i.) Resolva o problema usando programação dinâmica.

BOA PROVA!